Introduction of a model simulation program and its application in a demonstrative expert system to predict optimal spraying time for pest: PopModel 1.5 and Pest Forecast System

Kyung San Choi [1, 2]

[1] Research Institute of Climate Change and Agriculture, National Institute of Horticultural and Herbal Science, Rural Development Administration, Jeju 690-150, Korea

[2] Correspondence author, to: e-mail: mutant8@korea.kr

ABSTRACT

Three major frequently asked questions from farmers are about identification of the pest, an optimal pesticide spraying schedule, and the most effective pesticide for the targeted pest. Various protection calendars for each major crop have been provided. However, climate changes and the innovative development in the information techniques (IT_s) are uprising and making those protection calendars useless. Recent ITs expand in various field with a name of techniques, e.g., Bid data, Iot, AI, etc in order to automate machine, link information between machines, and help human's decision-making. However, the prediction of the pest occurrence and its damage in a crop is not easily done with these new techniques because the prediction is the new frontier that those techniques are employed to approach and overcome. More than three decades have been passed since the theoretical base of the pest model and algorithm for computing process on it were initiated. There is still no usable pest forecast system available for the growers mainly due to the lack of the techniques that can be easily simulated. I used a new concept for creating models with own calculation system and those are incorporated into a simulation program, Popmodel 1.5., with the simulation engine of PopModel 1.5. An expert system for managing pests are demonstrated with a program system called the Pest Forecast System. The Techniques in PopModel 1.5 and the visualized example of the Pest Forecast System will help open a new era in agriculture.

Keywords: model, simulation, protection, PopModel 1.5, pest forecast system.

INTRODUCTION

For thirteen years working in a local agricultural institute of a branch of government of Korea, I was often asked by farmers on a crop pest with the following three frequently asked questions (FAQs) which could be abbreviated into three words: What, When, and How. 'What' represents the identification of the pest. Most farmers have no interest in the scientific classification of the pest but whether the pest is controllable or not. 'When' refers to the optimal spraying time for effectively controlling one or several pests that occur at the same time. 'How' is usually about an effective control method. Farmers usually asked which pesticide is effective to control the pest and wanted the commercial name of pesticide sold in market. This preference on the commercial name may be one reason why a farmer continues to apply the same pesticide successively or frequently in a crop. They were confident with this particular one because it did not cause the pest's resistance problem based on their experience of using various pesticides. Protection calendars illustrating pests and their occurrence and damage season, a spraying schedule of the pesticides were provided and updated annually by a related agricultural station of the government and pesticide companies. However, such FAQs continued to emerge because the spraying schedules are different from farmer's cultivation methods and locations whereas the pest's occurrence and damage are dependent on the local climate and surrounding environment. Additionally, the identification of a pest and the selection of an effective pesticide for a pest(s) are not easily solved without the help of an expert.

Recently the greenhouse gases produced by human activity accelerate global warming (IPCC, 2014). The climate change causes immigration, accelerating occurrence time and shortening pests' overwintering period. Incursion of new pest, out-breaks of the pest, and abundant increase of the native pests will result in the economic loss in crops by destroying the pest management system based on a protection calendar. The information techniques (ITs) based on computer techniques have been developed and are expanding to various field with these new commercial brand names, e. g., Big-Data, IoT (Internet of Things), A.I. (Artificial Intelligence), etc. This kind of technique will help solve the identification of pest, provide recommendation of an optimal protection method, release farmers from labors, and so on by substituting the protection calendar. Therefore, automated machines and a convenient, reliable, and predictable

management system will be the best solution for answering those FAQs as well as coping with the dynamic situation caused by the climate change.

Prediction is a representative field and direction that the new ITs try to pursue and achieve. The effective protection period for the pest (i.e., spraying time) cannot be precisely predicted using the past collected and accumulated information in a database. For example, recent weather forecast has a high accuracy in weather prediction that was not achieved by using previously collected climate data in local weather station but by using simulation of various models. That is, the effective protection time for a pest or a group of pests that can be predicted by simulating the occurrence of the pests' models. Almost three decades have passed since Wagner et al. (1985) developed the programmatic simulation method for the insect model and Curry and Feldman (1987) founded a theory for the insect model. However, there have been a few articles validating their models through a simulation process, even though they are still not practical for farm usage as of today. The simple method to predict the insect occurrence is a Degree-Days-based model. Degree-Days-models have been widely used in disease, insect, and crop models because it is easy to build with a linear equation (Y=aX+b) and simple to apply. However, it couldn't tell when pests occur, but the peak time, so that all efforts failed to develop a sound decision-making system. While the combination of a developmental rate model (reviewed in Wagner et al., 1984a) and a developmental complete distribution model models (reviewed in Wagner et al., 1984b) build by a non-linear equation showed more reliable prediction capability, but this kind of models is hard to be used in practice without a help of a programmed program. These problems are basically caused from the lack of an appropriate program tool or a flat-form system to simulate the various models. Precisely predicable models are so complicated for being used in practice. They are composed of many stages or steps with a complicated linkage and various factors that affect the model. Furthermore, even a model for the same species, the model structure, content, and simulation method can be different from a model developer, the purpose of model, and application way. Although several commercial simulation programs (ie. STELLA (https://www.iseesystems.com), SIMILE (http://www.simulistics.com/), DYMEX (http://climatemodel.net/dymex.htm)) have been used in building and simulating models for crop, disease, and insect, The above-mentioned programs have a limitation in building such a complicated model so that the model built with those programs are mostly suitable for a study in a laboratory

level. There are many published papers for models but most models have no simulation of the model or have only low-grade simulation. Consequently, it is natural that models developed in the past have not been applied or failed in a farm level. In order to realize a future pest management system, techniques for building and simulating various kinds of models in insect, disease, and crop should be developed in advance.

With the goal of creating an omnipotent simulator that can build simply and simulate easily and fast in mind, various kinds models, a new simulation program was developed with the flat-forms, STAGE MODEL and POPULATION MODEL, for models with unique calculation system. PopModel1.5 is a latest version of the simulation program. This simulation engine is incorporated into a demonstrative expert system, Pest Forecast System (PFS),that can visualize a pest occurrence and the damage it causes in a crop and recommend some effective pesticides for the pest as well.

PopModel1.5

1. Program architecture

We used C/C++ to develop the system on function calculation, data management, and model simulation. In order to provide a user-friend interface, Windows API was used in coding PopModel1.5 and PMFList. PMFList is an add-on program to PopModel1.5 and provides a convenient environment for user to create, check, and manage functions that defined freely by user (Fig. 1). User can build and simulate freely the various models in the provided templates, STAGE MODLE and POPULATION MODEL with PopModel1.5 (Fig. 2). The STAGE MODEL is the unique and reliable template to contain any stage-type models that users think and is simulated reliably as standardized simulation (Fig. 2C). The POPULATION MODEL template is for a group of stages, which also shows a comprehensible structure of stages and their simulation procedure (Fig. 3). The data for forms of PopModel 1.5 is designed as a simple structure to be easily used in model simulation for the user's subjective usage. The user's data and the simulation outputs were displayed in a window and also transferred easily to Excel program and vice versa (Fig. 4).

Substantial times of tests were conducted by ensuring that the calculation codes and the simulation procedures in both templates, STAGE MODEL and POPULATION MODEL are error-free.

2. Calculation system

The data used in model simulation are a series of values. Thus, the values used in calculation are delivered or provided by other processes or code procedures, which causes difficulty in programing models and often results in a low efficiency and unexpected errors in calculation. However, our calculation system is based on the array concept. That is, each function can be calculated with the position information on "start", "now", and "end" in an array range and the position information can be used directly in the calculation of a function by using array variables, I, N, and T, respectively (Fig. 5). Therefore, user can access and use any valid position of the data. Also, our calculation system provides such various variables designated as P, X, R, C, Y, S, G for user's convenience. P variables are for the parameters used in a function (Fig. 1C). X and R variables are for user's data (Fig. 2B, 3C). For example, in a temperature-dependent

model, temperatures are used as the input for an X variable in the function. However, other data like density can be used directly or indirectly in that function. So, we provide R variable for those data. C variables are for the unit models composing a stage model because stage models are usually composed of several functions. So, the result of other functions or the formula of the function itself can be used in the simulation of a stage model. Those complicated process are easily done in our calculation system by using C variable (Fig. 2C). Y variable represents a formula and can be used as a variable indicating the output of the formula in a stage model (Fig. 2C). Our stage model can produce multi-outputs by one time simulation of the STAGE MODEL. S variables represent a stage that user sets to a stage in the population model and G variables are used for producing initial data of the stage model in the opulation model.

Our calculation system also provides such various defined functions as mathematical functions, condition functions, array functions for simulating various formulae and models.

3. STAGE MODEL

The stage model is so constructed to compensate different researchers' subjective or abstractive concept on the content and structure. For example, the number of functions and data in a stage model used among researchers are different and the simulation method/process can therefore be changed as for their practical application. In order to solve those problems, we figured out the stage model template to contain various models and to provide a confident simulation method (Fig. 2C). User can load functions in [COMPONENT FUNCTION] in the STAGE MODEL template and makes a procedure formula in [MAIN FUNCTION] using C variables indicating functions in [COMPONENT FUNCTION]. X or R variables used in [MAIN FUNCTION] indicate the data to be loaded in '[INPUT] X or R data' before simulation. The strong point of this stage model is that the original form of the functions loaded in [COMPONENT FUNCTION] are used without modifications. So, user can simulate any stage model with various data in a slight modification, if needed, of the formulae in a part of [MAIN FUNCTION], which helps comprehend model structure and process and also provide compatibility of the models between the model builder and the model user.

We provide three types of simulation method, CALCULATOR, ARRAY, and TRANSITION in [CAL. METHOD], which are a kind of simulation process and related

with the [CAL. RANGE] and [CAL. CONDITION] in the STAGE MODEL template (Fig. 2C). Because our calculation system is based on the array concept, the formulae in [CAL. CONDITION] and [MAIN FUNCTION] and each function in [COMPONENT FUNCTION] are calculated with the array information of "start", "now", and "end". User can also use the array information with the array variables, I, N, and T, respectively. CALCULATOR method produces the results according to the information in [CAL. RANGE]. In ARRAY method, the formulae in [MAIN FUNCTION] are calculated from the "start" to "end" position indicated in [CAL. RANGE] and produce results corresponding to the position in the array (Fig. 5A). [CAL. CONDITION] has an important role in ARRAY and TRANSITION (Fig. 5B, C). In case of the ARRAY with a starting condition, the formula in the 'Start if =' in the in [CAL. CONDITION] is calculated from the start to end indicated at [CAL. RANGE], and then the position value of I to be used in the calculation of the formulae in [MAIN FUNCTION] is set to the position where the result of the formula in 'Start if=' is true (=1) at first, thereafter the formulae in [MAIN FUNCTION] are calculated from the position of I to end producing results (Fig. 5B). TRANSITION method is an expanded form of ARRAY with 'Start if='. The simulation procedure of ARRAY with 'Start if=' is repeated whenever the result of the formula in 'Start if=' is true (= 1) for all the range indicated in [CAL. RANGE]. Finally, all the produced outputs from the formula in [MAIN FUNCTION] are summed according to their position in the array (Fig. 5C). That is, TRANSITION method is useful to simulate for the multi-cohort simulation.

Our stage model has a remarkable characteristic to produce more than one result by one time simulation. For example, stage emergence model produces only the output of the daily number of the group that completes their stage and go to the next stage. In order to know the remaining number of the group in the previous stage, we have to spend times to make a model and simulate it again. In PopModel1.5, user can easily get one or more results by adding one or more formula in [MAIN FUNCTION].

4. POPULATION MODEL

The POPULATION MODEL template of PopModel1.5 is designed for the group of stages with a various linkage and simulation procedure (Fig. 4A). Stages are simulated in the easy and comprehensible order from top to bottom and from left to right following the arrow from the start stage determined by user. Users do not need to

make model again in order to change the starting stage but only move the indication mark of the start stage to other stage that users want to start at first. Also users determine which stage should be re-entered, if needed, by setting a Loop arrow and determine the number of re-entrance. Popmodel1.5 visualized automatically the simulation order of stages from the start stage with arrows. Therefore, any users comprehend easily the whole progress in the simulation of any population models. In a STAGE MODEL in the POPULATION MODEL template, users can use other outputs in the other stage by using S variables after being set to the stage. S0 is a variable reserved for the initial data in each stage and can be used, if it is necessary to simulate the stage model at the first simulation (Fig. 4D). The simulation procedure of the stage model in population is a little difference from the stage model simulation. As well as the data loaded X, R, and S variable itself, a new data produced through a calculation with those data in the simulation of a stage. So, we provide G variable that represents the data produced by calculating the formula described in 'G var' in the STAGE MODEL template. Therefore, the data for G variable was produced at first and then STAGE MODEL is simulated as the procedure mentioned in STAGE MODEL section according to the information that user inputs.

POP ULATION TEMPLATE provides three sections: PRERUN, [MAIN], POSTRUN. PRERUN is for a section when user needs to produce data that is necessary for the [MAIN]. POSTRUN is for a section where user arranges the output that user really needs from the previous sections, PRERUN and [MAIN].

Pest Forecast System

1. Program architecture

The Pest Forecast System comprises the following two programs, PFSmanger and PFSviewer. They are programed with C/C++ and Windows API. PFSmanager is designed for a manager in the level of a Server where all the basic information and models for the crop, insect, disease, and agrochemical could be managed and updated. PFSviewer is a client program for a farmer who can get the predicted an optimal spraying time and best pesticide with the information that farmer selected the crop in and location of a farm. The meteorological date required for simulation of the prediction models for the pest is designed automatically accumulated by a program, CMDBmanager, equipped in the Pest Forecast System. This system is a demonstration program for developing an expert system to be used in agriculture in future.

2. Meteorological data

The CMDBmanger in the Pest Forecast System is designed to start checking for new weather data and collecting them in its own Database whenever the models of the pest that should be simulated according to the setting information by farmer. The CMDB collects the daily weather information of the location by visiting the website of the Korean meteorological administration (KMA, http://www.kma.go.kr/). In this program, the collected weather factors are low temperature, high temperature, average temperature, humidity, precipitation, amount of snowfall, and wind. The normal weather data provided by the KMA are also collected as well as the daily data. The Collected data are accumulated in its own database. In order to predict a future, data like a pest occurrence and crop growth needs basically a future weather information, which is provided by filling the normal weather data after the last accumulated weather data in the database. With this process, the weather data for simulating the pest occurrence is automatically provided in PFSviewer.

3. PFSmanager

Most role of PFSmanager is to manage the five databases named as ‘Refer DB’, ‘Climate DB’, ‘AgroChem DB’, ‘Crop Pestlist’, and ‘Model DB’ (Fig. 8A). The content of each DB is visualized in the right window as for the option (Fig. 8B, 8C). ‘Refer DB’

is for the classification of the crop, disease, insect, and agro-chemicals, which is a basic information used commonly in 'AgroChem DB', 'Crop Pestlist', and 'Model DB'. Even in the same pest, the status and the importance for the control are different among crops. The database for 'Crop Pestlist' is a basic information used for providing the optimal spraying time and recommending the best pesticide. The database for 'Agrochem DB' is for the legal agro-chemicals. This information is basically used in recommending the best pesticide, too. The function of recommending pesticides for the pest in a crop is provided in the option of 'Agrochem DB'. The 'Model DB' is for the model for insect, disease, agro-chemical, and crop, but only insect model is working in this version. This version of the Pest Forecast System is a program for a demonstration to visualize what kinds of Database and information are required and how they are to be managed at least.

4. PFSviewer

PFSveiwer is a client program for a farmer. This kind of program should be simple to use and deliver useful and compacted information to a farmer with a comprehensible form. Farmer only needs to set the location of a farm and the crop in the farm (Fig. 9A). The predicted level of the occurrence and damage of the pest in a crop is visualized with blue and red line, respectively in a year (Fig. 9C). This program provides a diary to record the history of the applied pesticide information in 'Agrochem DB' (Fig. 9D). Those applied information is visualized in Fig. 9C with the pest occurrence and damage information as related to the date with an effective level of a pesticide to each pest. Predicted spraying time and the recommended pesticide is shown in Fig. E. PFSviewer is a demonstration program to show what kinds of information should be delivered for a farmer, how the information are to be visualized.

ACKNOWLEDGEMENTS

PopModel1.5 is an upgraded version from PopModel1.0 and the copyrights were reserved for Choi, Kyung San and Kim, Dong-Soon in 2017 (C-2017-003991). The copyrights of CMDBmanager is reserved by RDA (C-2017-012553) and that of Pest Forecast System is reserved by RDA, Jeju national university, Andong national university, and GyeongSang national university (C-2017-030630). All the above-mentioned programs were developed by Choi, Kyung San. RDA owns the right on two patents (10-1578421 and 10-1589235) for the STAGE MODEL and its calculation system of PopModel1.5 and a Korean company, Jaemi, got an exclusive license by December of 2017.

LITERATURE CITED

Choi, K. S. and Kim, D. S. 2014. Temperature-dependent development of *Ascotis selenaria* (Lepidoptera: Geometridae) and its stage emergence models with field validation. Crop Protection. 66:72-79.

Choi, K. S. and Kim, D. S. 2016. Effect of Temperature on the fecundity and longevity of *Ascotis selenaria* (Lepidoptera: Geometridae): developing an oviposition model. J. Eco. Entomol. 109:1267–1272.

Curry, G. L. and Feldman, R. M., 1987. Mathematical Foundations of Population Dynamics. In: Monograph Series 3. Texas Engineering Experiment Station, College Station, TX.

IPCC (2013). Annex II: Climate system scenario tables. Climate Change 2013: The physical science basis. Contribution of working group I to the fifth assessment report of the Intergovernmental panel on climate change (ed. by T. F. Stocker, D. Qin, G. -K. Plattner, M. Tignor, S. K. Allen, J. Boschung, A. Nauels, Y. Xia, V. Bex, and P. M. Midgley), pp. 1395-1445. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA.

Wagner, T. L., Wu, H., Sharpe, P. J. H., Schoolfield, R. M., and Coulson, R. N. 1984a. Modeling insect development rate: a literature review and application of a biophysical model. Ann. Entomol. Soc. Am. 77:208-225.

Wagner, T. L., Wu, H., Sharpe, P. J. H., and Coulson, R. N. 1984b. Modeling distribution of insect development time: a literature review and application of Weibull function. Ann. Entomol. Soc. Am. 77:475-487.

Wagner, T. L., Wu, H., Feldman, R. M., Sharpe, P. J. H., and Coulson, R. N., 1985. Multiple cohort approach for simulation development of insect population under variable temperatures. Ann. Entomol. Soc. Am. 78:691-704.

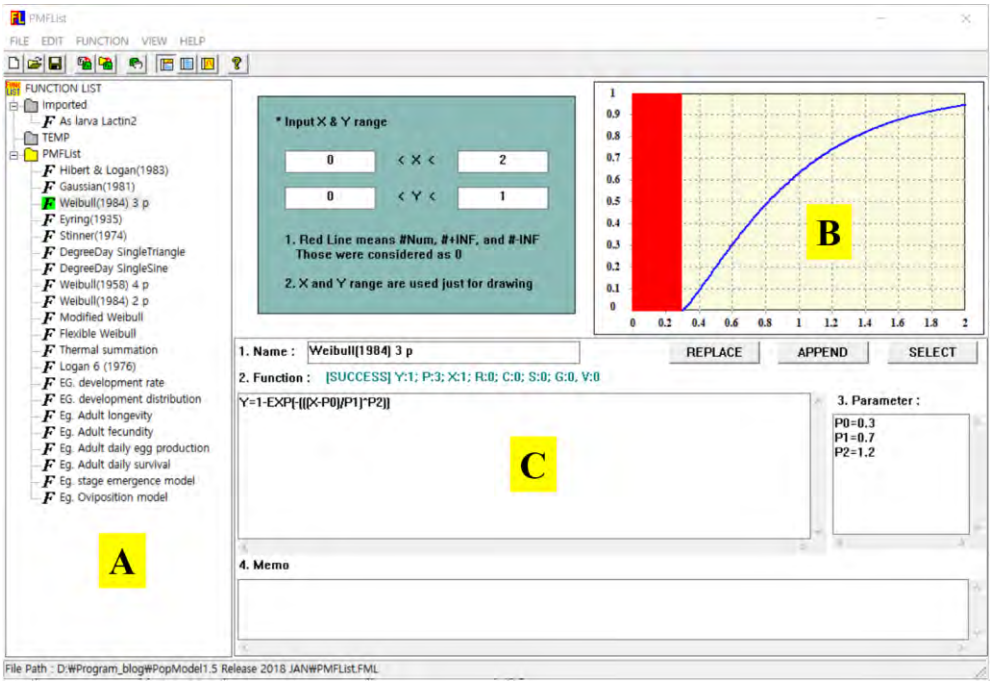

Fig. 1. Screen of the PMFList program. (A) Files containing the functions are displayed, (B) The figure of function selected by user, and (C) Function information is displayed and can be edited by the user.

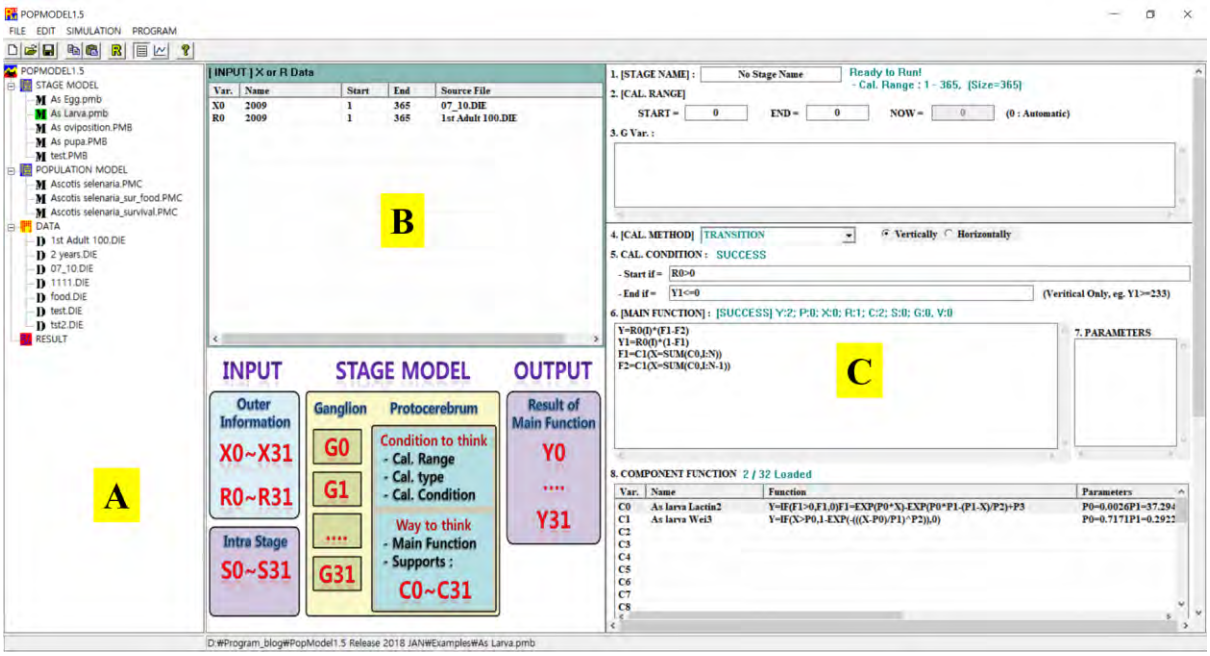

Fig. 2. Stage model. Screen of PopModel1.5 program. (A) Files used in PopModel program, (B) Data loaded on X or R variable are used in the simulation of the stage model, and (C) The information of a stage model.

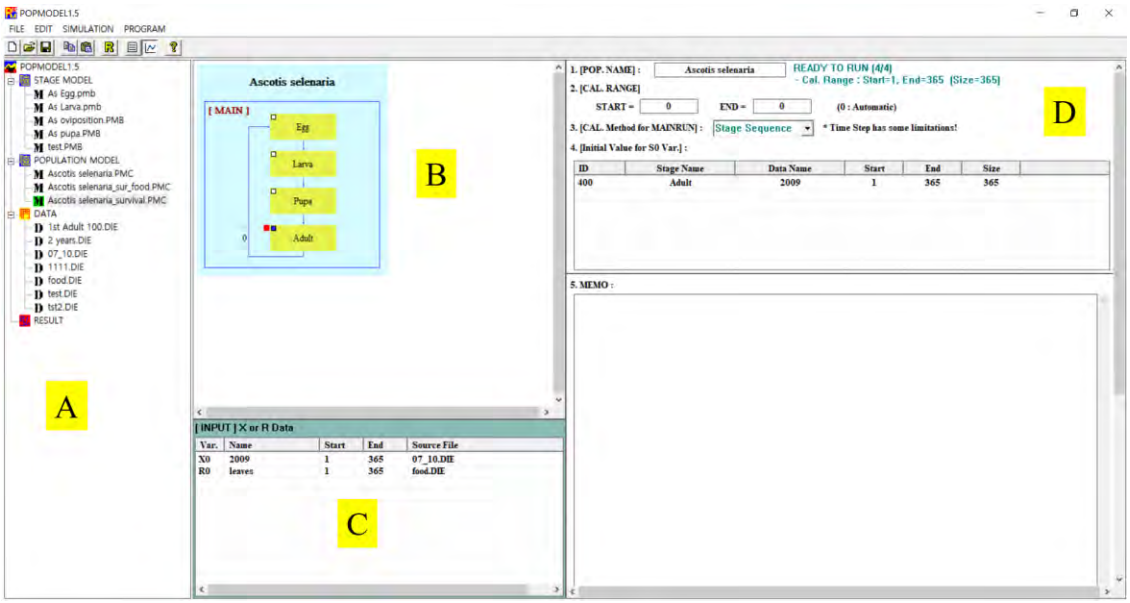

Fig. 3. Population model. Screen of PopModel1.5 program. (A) Files used in PopModel program, (B) GUI Builder for a population model with stages and their linkage, (C) Data loaded on X or R variable are used in the simulation of the population model, and (D) the setting window for the population model.

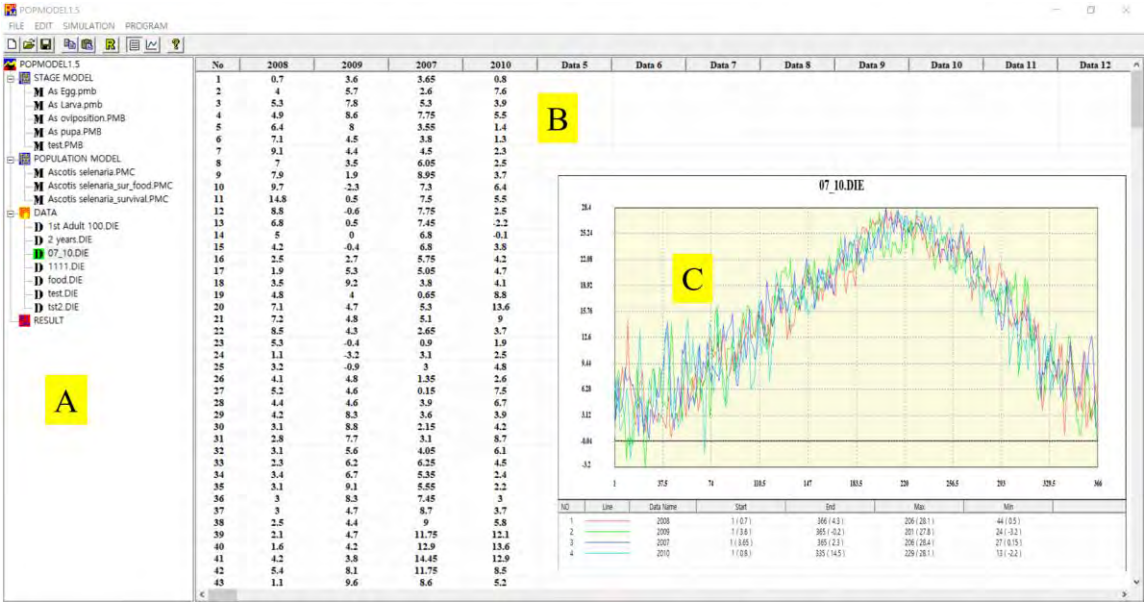

Fig. 4. Data Screen of PopModel1.5 program. (A) Files used in PopModel program, (B) and (C) Data in a file are displayed in text mode or graph mode, respectively.

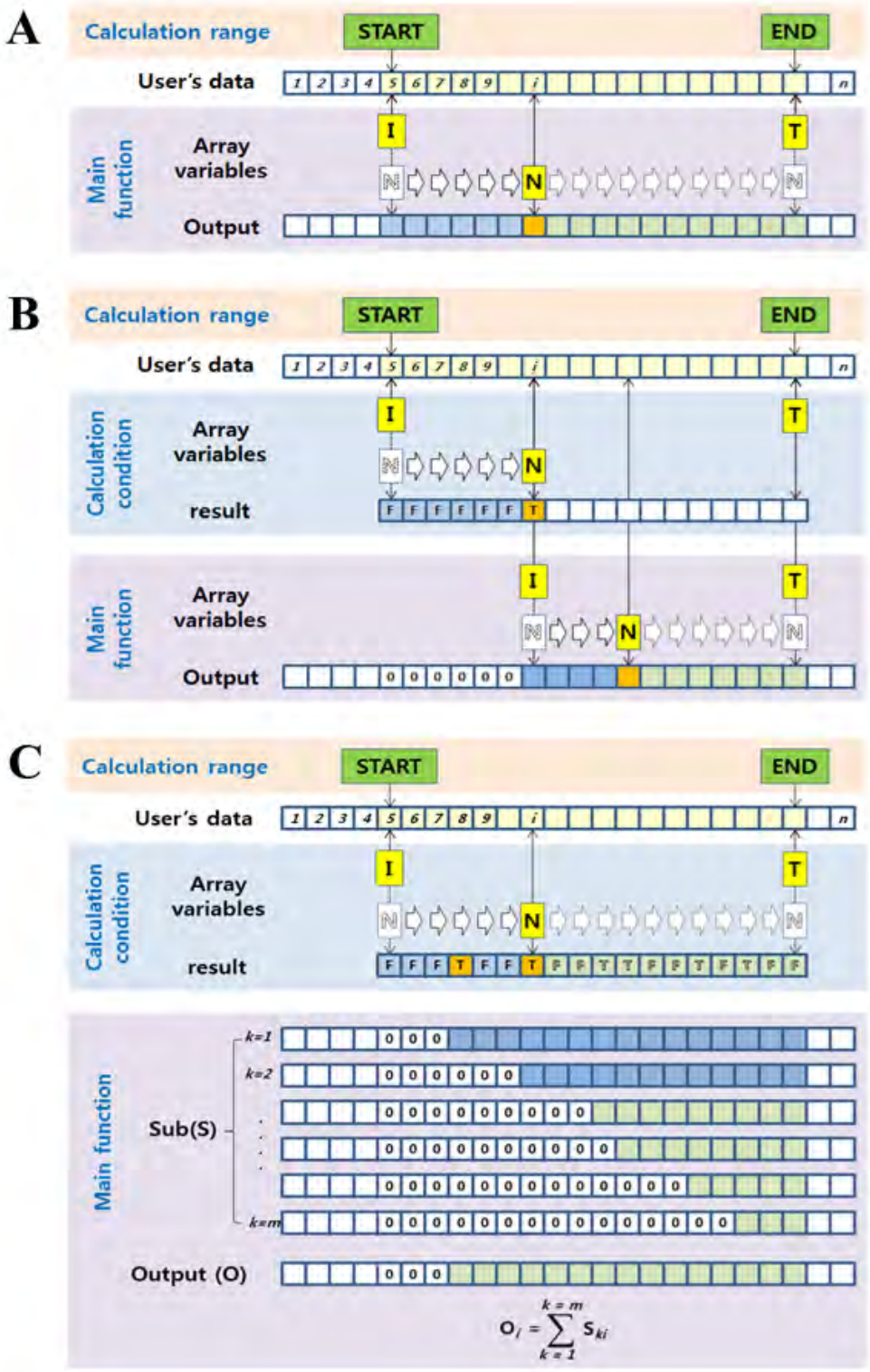

Fig. 5. Scheme of calculation method of the stage model in PopModel1.5 program. (A) Array, (B) Array with a conditioned start, and (C) Transition.

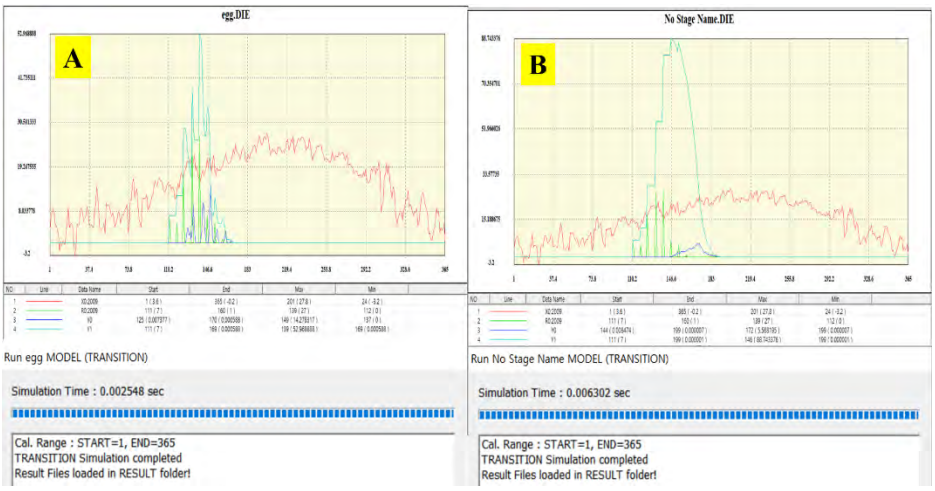

Fig. 6. Outputs of the stage models simulated with PopModel1.5 program according to the daily temperature data (red line). (A) The occurrence of the hatched larva (blue line) and the remaining egg population (light blue line) were predicted from the egg laid data (green line) with the egg emergence stage model (B) The pupal occurrence (blue line) and the remaining larval population (light blue line) were predicted from the predicted hatched larva data (green line) with the larval emergence stage model. Stage emergence models of *Ascotis selenaria* and their simulation method were described in Choi and Kim (2014).

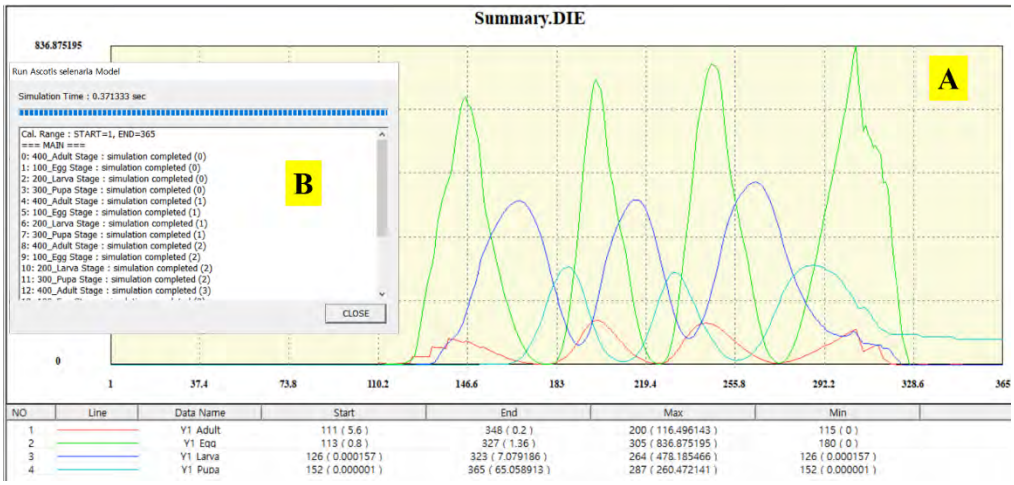
Summary.DIE
836.875195
Run Ascotis selenaria Model
Simulation Time : 0.371333 sec
Cal. Range : START=1, END=365
=== MAIN ===
0: 400_Adult Stage : simulation completed (0)
1: 100_Egg Stage : simulation completed (0)
2: 200_Larva Stage : simulation completed (0)
3: 300_Pupa Stage : simulation completed (0)
4: 400_Adult Stage : simulation completed (1)
5: 100_Egg Stage : simulation completed (1)
6: 200_Larva Stage : simulation completed (1)
7: 300_Pupa Stage : simulation completed (1)
8: 400_Adult Stage : simulation completed (2)
9: 100_Egg Stage : simulation completed (2)
10: 200_Larva Stage : simulation completed (2)
11: 300_Pupa Stage : simulation completed (2)
12: 400_Adult Stage : simulation completed (3)
CLOSE
B
A
0
1
37.4
73.8
110.2
146.6
183
219.4
255.8
292.2
328.6
365
NO
Line
Data Name
Start
End
Max
Min
1 Y1 Adult 111 (5.6) 348 (0.2) 200 (116.496143) 115 (0)
2 Y1 Egg 113 (0.8) 327 (1.36) 305 (836.875195) 180 (0)
3 Y1 Larva 126 (0.000157) 323 (7.079186) 264 (478.185466) 126 (0.000157)
4 Y1 Pupa 152 (0.000001) 365 (65.058913) 287 (260.472141) 152 (0.000001)

Fig. 7. Outputs of the population model of *Ascotis selenaria* simulated with PopModel1.5 program. (A) 1st to 4th occurrence of the generation at each stage; egg (red line), larva (blue line), pupa (light blue line), and adult (green line) (B) Simulation progress window for the population model. Stage emergence models and adult's oviposition model of *Ascotis selenaria* and their simulation method were described in Choi and Kim (2014, 2016).

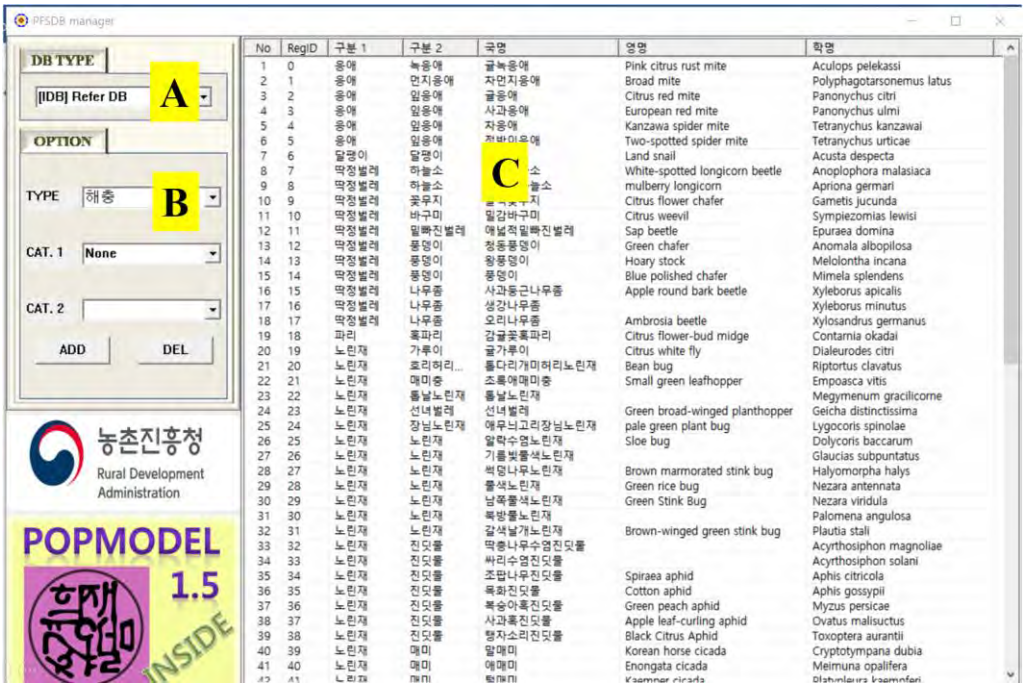
PFSDB manager
DB TYPE
[IDB] Refer DB
A
OPTION
TYPE
해충
B
CAT. 1
None
CAT. 2
ADD
DEL
농촌진흥청
Rural Development Administration
POPMODEL
1.5
INSIDE
C
No RegID 구분 1 구분 2 국명 영명 학명
1 0 응애 녹응애 귤녹응애 Pink citrus rust mite Aculops pelekassi
2 1 응애 먼지응애 차먼지응애 Broad mite Polyphagotarsonemus latus
3 2 응애 잎응애 귤응애 Citrus red mite Panonychus citri
4 3 응애 잎응애 사과응애 European red mite Panonychus ulmi
5 4 응애 잎응애 차응애 Kanzawa spider mite Tetranychus kanzawai
6 5 응애 잎응애 점박이응애 Two-spotted spider mite Tetranychus urticae
7 6 달팽이 달팽이 Land snail Acusta despecta
8 7 딱정벌레 하늘소 White-spotted longicorn beetle Anoplophora malasiaca
9 8 딱정벌레 하늘소 mulberry longicorn Apriona germari
10 9 딱정벌레 꽃무지 Citrus flower chafer Gametis jucunda
11 10 딱정벌레 바구미 Citrus weevil Sympiezomias lewisi
12 11 딱정벌레 밀빠진벌레 애넓적밀빠진벌레 Sap beetle Epuraea domina
13 12 딱정벌레 풍뎅이 청동풍뎅이 Green chafer Anomala albopilosa
14 13 딱정벌레 풍뎅이 왕풍뎅이 Hoary stock Melolontha incana
15 14 딱정벌레 풍뎅이 풍뎅이 Blue polished chafer Mimela splendens
16 15 딱정벌레 나무좀 사과둥근나무좀 Apple round bark beetle Xyleborus apicalis
17 16 딱정벌레 나무좀 생강나무좀 Xyleborus minutus
18 17 딱정벌레 나무좀 오리나무좀 Ambrosia beetle Xylosandrus germanus
19 18 파리 혹파리 감귤꽃혹파리 Citrus flower-bud midge Contarnia okadai
20 19 노린재 가루이 귤가루이 Citrus white fly Dialeurodes citri
21 20 노린재 호리허리... 톱다리개미허리노린재 Bean bug Riptortus clavatus
22 21 노린재 매미충 조록애매미충 Small green leafhopper Empoasca vitis
23 22 노린재 톱날노린재 톱날노린재 Megymenum gracilicorne
24 23 노린재 선녀벌레 선녀벌레 Green broad-winged planthopper Geicha distinctissima
25 24 노린재 장님노린재 애무늬고리장님노린재 pale green plant bug Lygocoris spinolae
26 25 노린재 노린재 알락수염노린재 Sloe bug Dolycoris baccarum
27 26 노린재 노린재 기름빛물색노린재 Glaucias subpuntatus
28 27 노린재 노린재 썩덩나무노린재 Brown marmorated stink bug Halyomorpha halys
29 28 노린재 노린재 풀색노린재 Green rice bug Nezara antennata
30 29 노린재 노린재 남쪽풀색노린재 Green Stink Bug Nezara viridula
31 30 노린재 노린재 톱방패노린재 Palomena angulosa
32 31 노린재 노린재 갈색날개노린재 Brown-winged green stink bug Plautia stali
33 32 노린재 진딧물 딱총나무수염진딧물 Acyrthosiphon magnoliae
34 33 노린재 진딧물 싸리수염진딧물 Acyrthosiphon solani
35 34 노린재 진딧물 조팝나무진딧물 Spiraea aphid Aphis citricola
36 35 노린재 진딧물 목화진딧물 Cotton aphid Aphis gossypii
37 36 노린재 진딧물 복숭아혹진딧물 Green peach aphid Myzus persicae
38 37 노린재 진딧물 사과혹진딧물 Apple leaf-curling aphid Ovatus malisuctus
39 38 노린재 진딧물 탱자소리진딧물 Black Citrus Aphid Toxoptera aurantii
40 39 노린재 매미 말매미 Korean horse cicada Cryptotympana dubia
41 40 노린재 매미 애매미 Enongata cicada Meimuna opalifera

Fig. 8. Screen of PFSmanager of the Pest Forecast System. (A) Database types used in PFSmanager, (B) Options for each database, and (C) Content of the database.

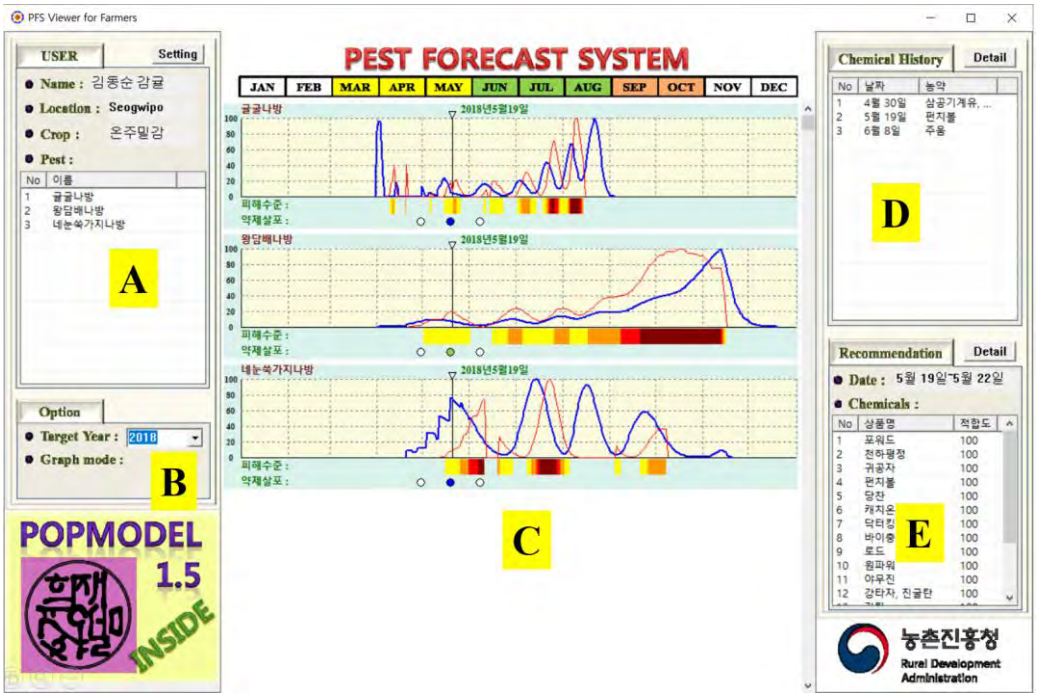

Fig. 9. Screen of PFSviewer of the Pest Forecast System. (A) Setting information of a farmer, (B) Year to be simulated, (C) Visualize the pest occurrence, spraying time, and the effectiveness of pesticide at the sprayed time the selected year, (D) History of applied chemical information at a crop in a location by the farmer, and (E) Optimal spraying period to be controlled and the recommended pesticide with their profitability to the pest to be controlled.